Bodacious Battle Bots

USER’S

MANUAL

User’s Manual

Table of Contents

1. GENERAL INFORMATION ............................................................................................2

Overview .........................................................................................................................3

1. STARTING OUT…............................................................................................................4

Navigating Misty Robotics Website ...............................................................................5

Charging Misty …………………………………………………………………………9

Getting Misty IP address ……………………………………………………………... 11

Generating JSON for Skills …………………………………………………………...12

Running Skills .............................................................................................................. 14

1. **GENERAL INFORMATION**

**OVERVIEW**

**Project Details:**

Ian Bernstein is the founder of Misty Robotics. He created a robot alongside a platform that just about anyone can use. The robot has uses for business, education, research, and personal. Ian built Misty with developers in mind. Misty has a range of tool sets to accommodate developers in building their own skills. Misty is quite adaptable in the sense that she may extend her functionality by use of APIS, hardware modifications, and additional sensors.

The Bodacious Battle Bots were tasked with utilizing an API that supports communication between multiple Misty robots. The Battle Bots worked alongside the Misty Mountain team to achieve our collaborative tasks. The main task that both teams had to work together to complete is the translation. One Misty bot is tasked with listening to audio (namely human) and translating it into another language. The base language our robot will except is English which is translated into Spanish. Misty Mountain’s robot will output a similar result after the Battle Bots robot verbally speaks the transcribed text into the new language (Spanish). Misty Mountain’s robot will then take the Spanish text and translate it into another set language and play the audio in yet another language. Misty Mountain’s robot response will depend on what the Bodacious Battle Bots robot says after translating the original statement. Some tasks utilize our built from scratch dashboard to imitate communication between multiple robots. Since we couldn’t utilize web sockets for the robot to directly communicate via Ip addresses, each robot will SEND, POST, and GET data from the dashboard for collaborative tasks. The dashboard will be used to connect to individual robots using their IPs. Skills will be run directly from the dashboard because each skill will be independently loaded on each robot.

Overall, these tasks demonstrate some of the capabilities of the Misty robot while simultaneously providing educational value.

**About the team:**

The Misty Robot Collaboration could only be made possible by the hard work of the Bodacious Battle Bots development team; members include Pablo Boudet, Luke Branson, Anikko Barton, Gabi Goncalves, and Henrique Medeiros. Our leader Henrique made sure that all members of our team adequately contributed to the project within their respected roles. Luke and Pablo primarily worked on the GUI and functionality of our dashboard. Gabi, Anikko, and Henrique provided much needed research and created multiple skills for Misty. All members of the team also pitched in for documentation since there were multiple segments to the project.

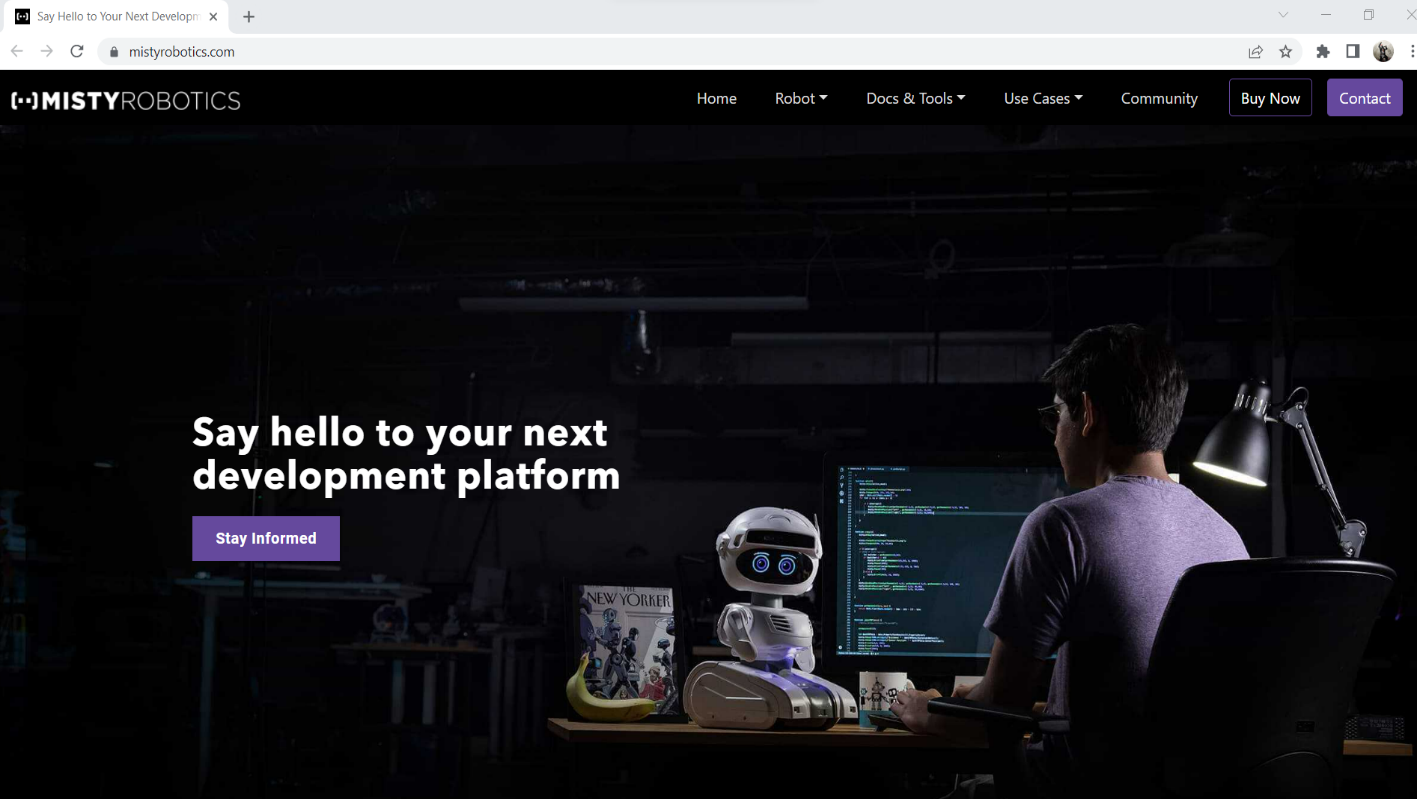
1. **STARTING OUT**

**Starting Out**

Navigating Misty Robotics Website

1. Before starting your journey developing your own skills to run on the Misty robot, you should familiarize yourself with the Misty Robotics website. First access a web browser of your choice. We used Google Chrome and Microsoft Edge. Access the website at the URL: <https://www.mistyrobotics.com>

The front page will look like this (skip to page 9 if you are already familiar with the site)

From the homepage you may quickly access several key tabs. The robot tab contains two pages including Misty II which iterates her capabilities. The accessories page simply displays available attachments for misty which further extend what she is capable of.

Docs & Tools contains the Technical Overview, Dev Tools, Documentation, and Misty Labs.

The Use Cases includes multiple practical implementations of Misty including Retail, Eldercare, Innovation Labs, education and research, and the use case overview.

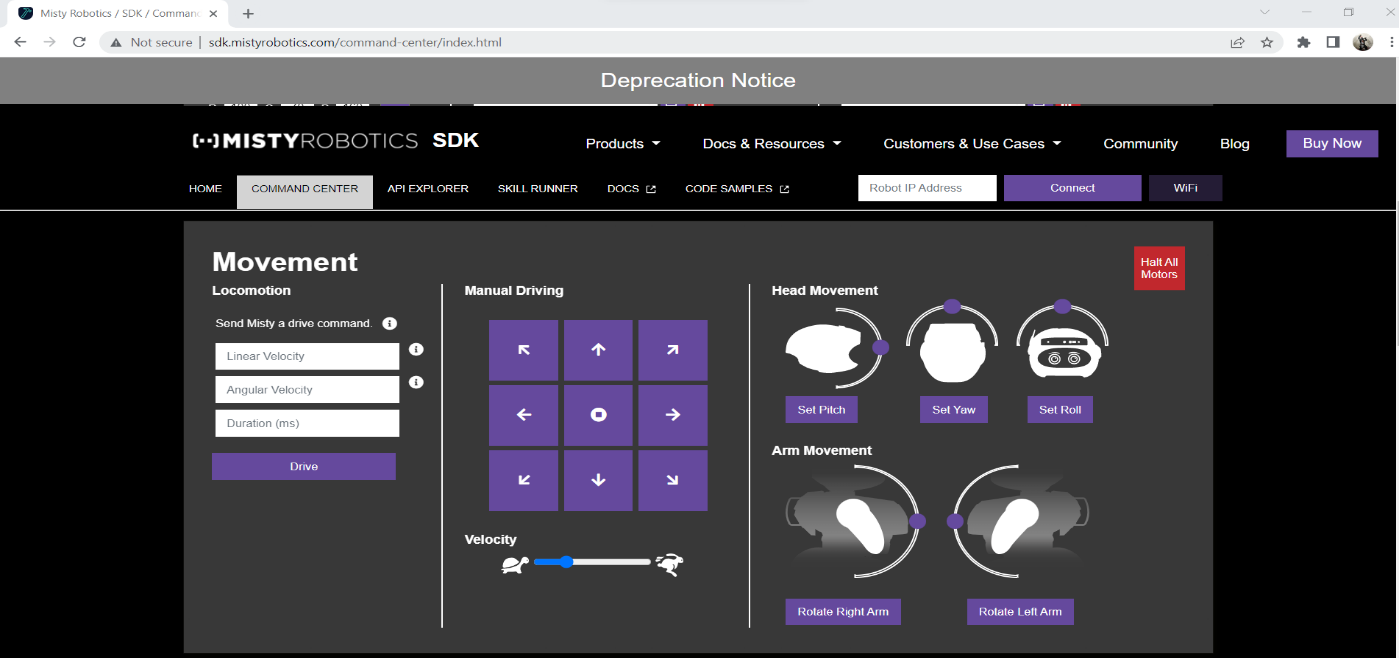
The community tab redirects the user to a forum which not only patch updates and release notes, but also original postings regarding skill construction by prospective developers. You may find help here if someone has had a similar issue.

Buy Now is simply where you can purchase Misty robots and accessories through the website.

You may contact the developers of Misty directly through the contact tab or the forums.

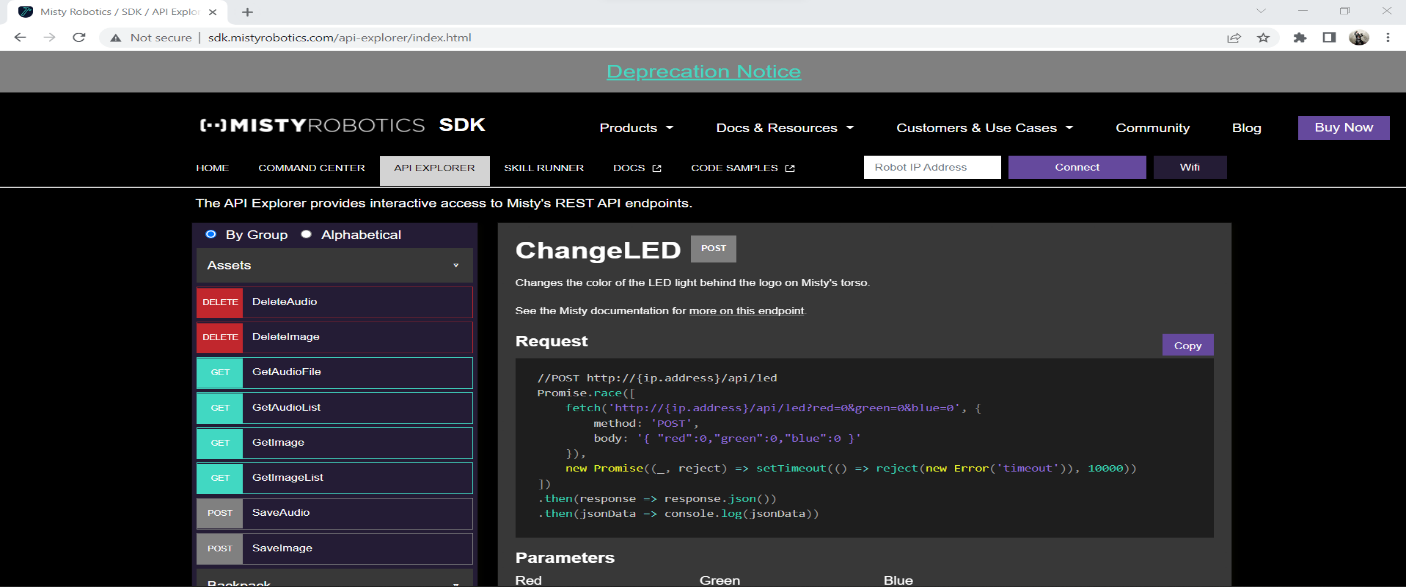
You may use the command center to manually run basic functions of Misty manually without writing any code. This includes connecting the robot using its associated IP address. See step 4 for obtaining the robot's IP address.

Access command center by first clicking the Docs & Tools tab and selecting Dev Tools.



Select the command center tab you see next to home and API explorer.

The API Explorer tab displays REST API endpoints that can be sent as commands to Misty.

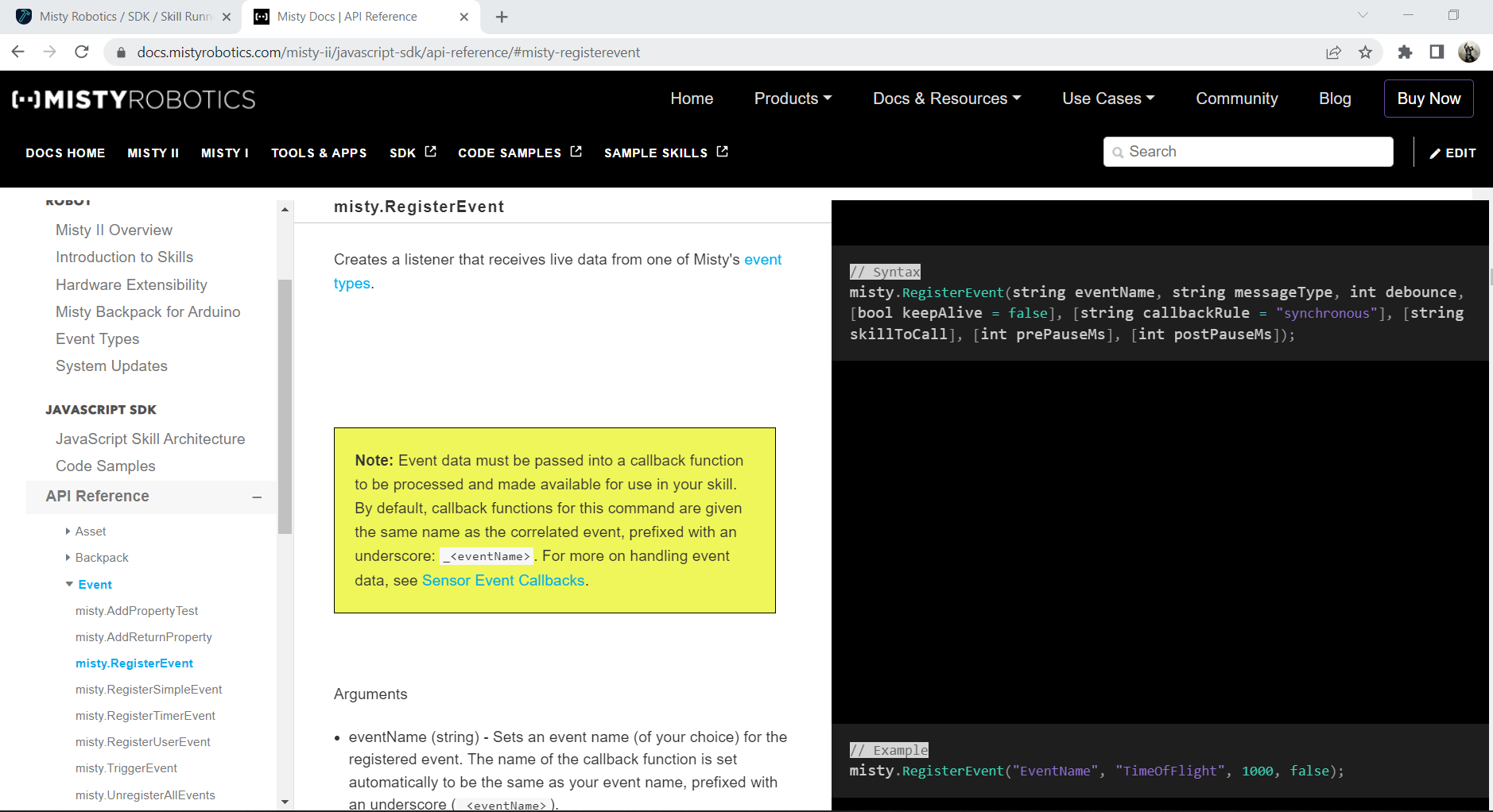


Although the REST API is fully suitable while working in conjunction with web sockets for executing commands with Misty, the Bodacious Battle Bots development team chose to employ JavaScript API and skill runner to execute skills. (More detail on skill runner section of manual)

From here you may access the docs page on the tab to the right of the skill runner or select documentation from the Docs & tools dropdown.

You may also find documentation regarding the REST API and .NET beta here.

Use this documentation to assist you in developing your skills! If you need more assistance or a boost, check out the code samples on Github. (Code samples tab)



Misty’s battery tends to drain quickly so you need to keep her charged.

* Simply plug in the charging dock and place misty on top of it (see below photos)

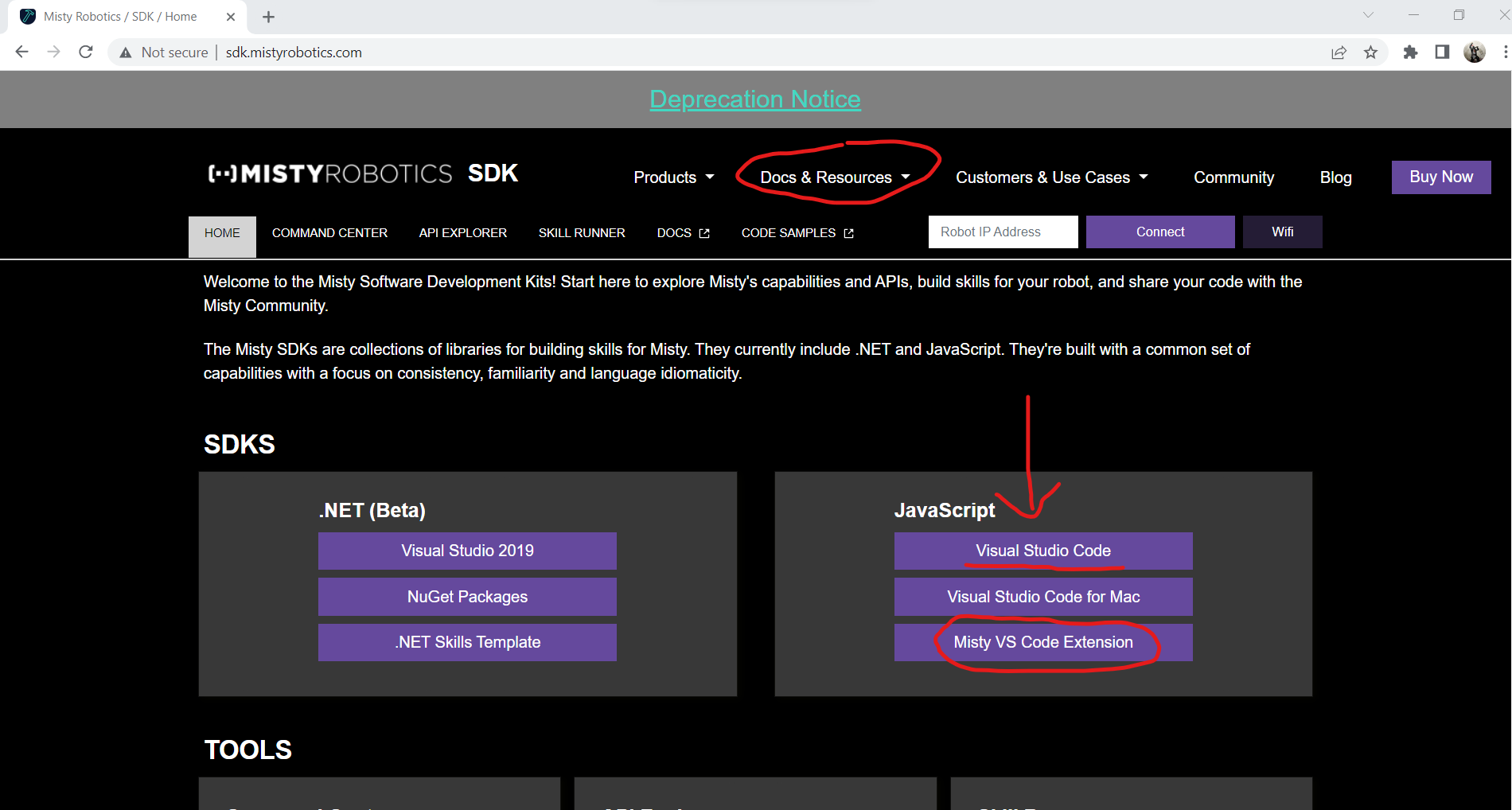
A picture containing indoor

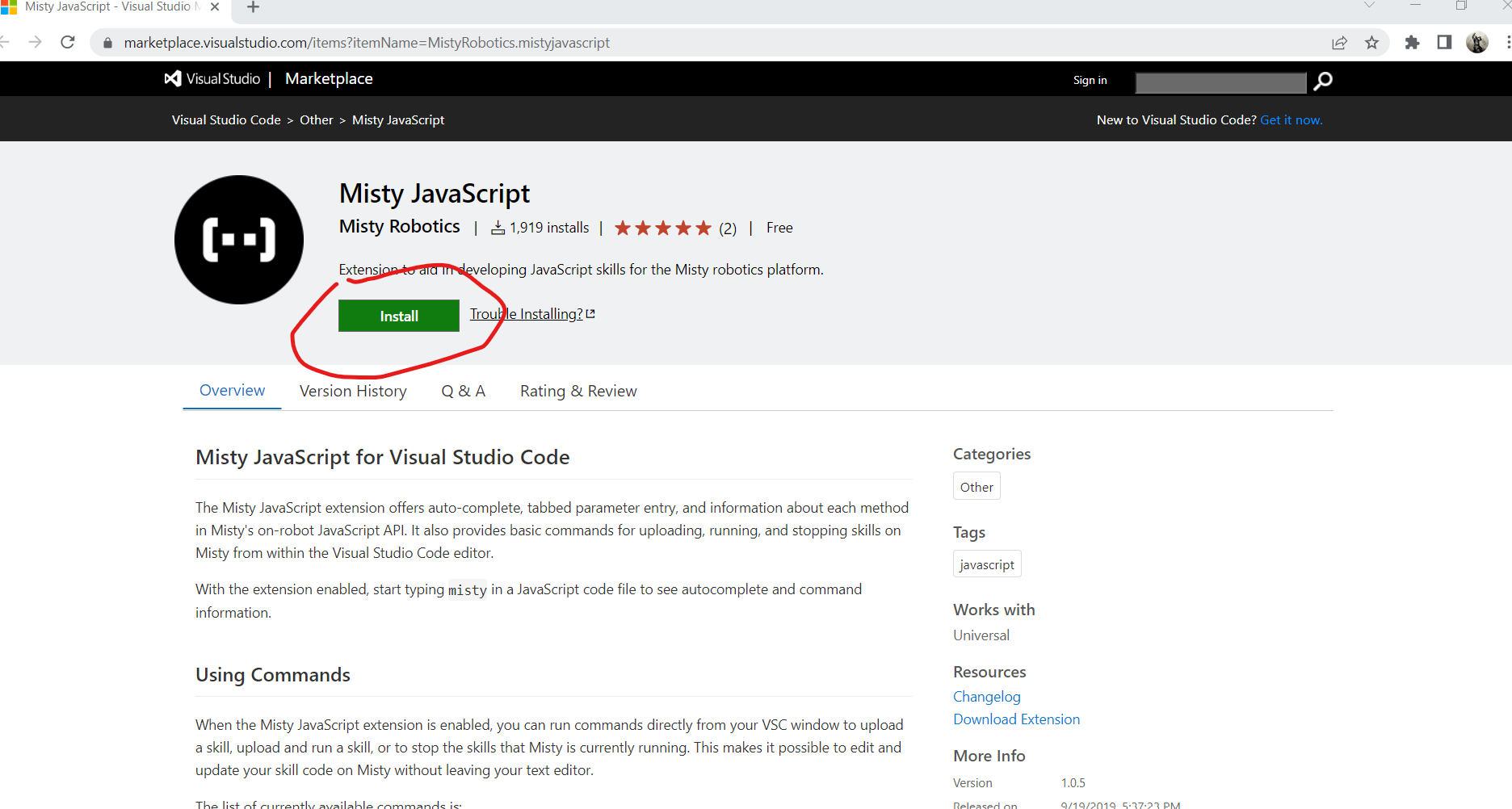
Description automatically generatedA picture containing indoor

Description automatically generated

You may keep Misty powered on while she charges, however, we suggest charging her while she is powered off. You may continue running and loading skills if she is powered on while charging. If she is charging you should hear a fan of some sorts running.

2. Before you start coding you should install SDKs for the Misty Library. As you can see in the picture below, the SDKs available for use with the extension are Visual Studio 2019 for .NET(Beta) and Visual Studio Code for JavaScript. The team primarily used Visual Studio Code to develop skills using the JavaScript programming language. We simply used it because it has more support and less bugs.

Simply install the extension after you **FIRST** install Visual Studio Code if you haven’t already.



3. Obtain the IP address of Misty directly through the Misty App.

-Connect to LU-Robotics wi-fi on mobile device (you may need to get password from Dr.Jeevanjee)

-Install the app from the App Store or Play Store

Graphical user interface, application

Description automatically generated

- Turn Misty on. Her power switch is located underneath her towards her backside. A few animations will play on her display while she loads. When her eyes are open wide, she should be ready to connect.

- Connect to the Misty robot using Bluetooth on your mobile device

Graphical user interface, text, application

Description automatically generated A screenshot of a computer

Description automatically generated with low confidence

-The IP address and battery life will display shortly after connecting to Misty

4. Generating JSON and JS file. Click Skill Runner:

Graphical user interface, website

Description automatically generated



Under Generate, type the skill name and select download.

Graphical user interface, website

Description automatically generated



Lastly, click the “Generate JSON Meta Template” and save it.

For the Javascript file, simply name it **EXACTLY** the same as you did for your JSON and make sure that you have the .js extension.

Example skillname and its JSON: DoTheTask.js works with DoTheTask.json (Correct because they are identical). Dothetasks.json(wrong because it isn’t identical to the name of the js file).

Text

Description automatically generated

Above is one of the skills our team developed so that you may see the JSON file it uses. TicTacToe under “Name” is the same as the TicTacToe.js file name. The name that will appear there is the one you generated. This can be updated within visual studio if need be. You may generate a unique ID using a GUID generator or what we used at URL: <https://www.guidgenerator.com/online-guid-generator.aspx>

The website will not except the js files without their accompanying json files!

You should also make sure your js and json files and any other files that coincide with a particular skill are all within the same folder.

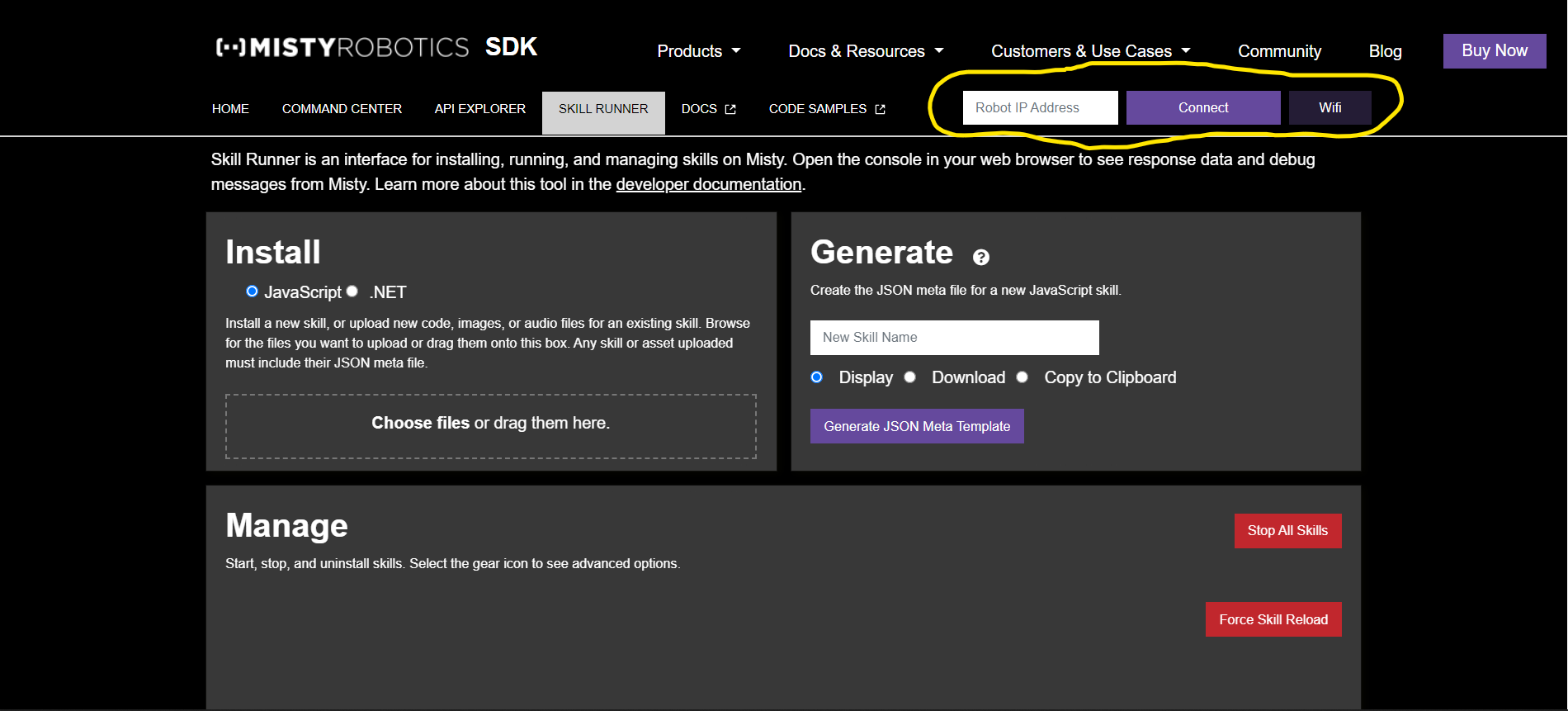
**Starting Out**

Running Skills

1. Make sure you connect to the LU-Robotics or Lindenwood Wi-Fi on your computer before continuing. You may need to contact Dr. J to gain access to the password for LU-Robotics. Make sure you have Misty turned on.

2. Once you’re connected, select the skill runner tab under the development tools dropdown like you did for the command center.

3. Input the Ip address found through the mobile app (reference pg. 11) in the top right portion of the page where you see Robot Ip address near the connect button. Then click connect. If you have an issue connecting refresh the page and turn the robot off, then back on.

4. Either click or drag your JavaScript, image, audio, and JSON files where you see in rectangular box” Choose files or drag them here”. The website will not load the files if it does not have an accompanying JSON file! You may generate a basic one to the right of the install option. (Reference pages 12-13 for generating JSON)

5. If your skills and files load successfully into Misty, you will see them appear under Manage. This is where you may run, stop, reload, and delete skills currently on Misty. When loading new skills into Misty it is good practice to force skill reloads when loading new skills. The site does allow you to upload A skill with the same name, but you should delete the skill before uploading it once more.

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

As you see above, after your skills load successfully, they will appear under manage. From here you may delete individual tasks using the trash bin button. As stated previously, if you already have a skill loaded onto Misty with the same skillname, delete it and force skill reload before you upload the updated files.

Although it isn’t required for you to use the debugger when running skills, we highly suggest you use it in order to debug your own code and visually see what code is executing and at what times.

Simply click the start button to run a specific task. Click stop to cancel a skill or stop all skills if multiple is running.

A white robot in front of a computer

Description automatically generated with low confidence

Here you can see our Dance skill running successfully on Misty.

Now that you understand how to run the skills, let’s look at what happens when your skills run and stop in the debugger.

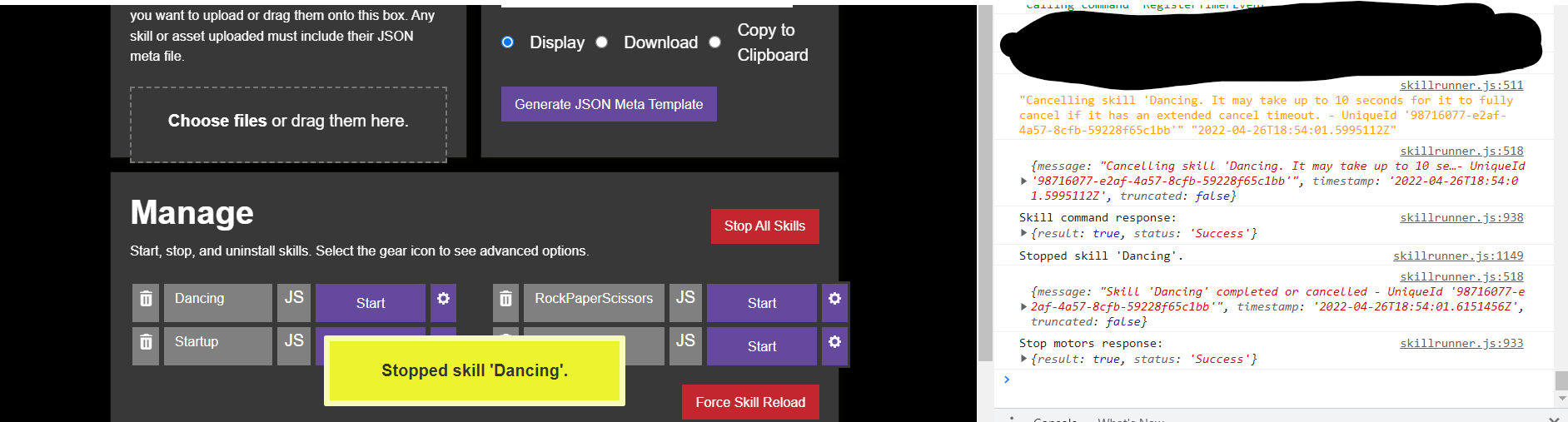
The photo below shows the beginning contents of the console after the Dance skill executes.

A picture containing text

Description automatically generated

6. Right click somewhere on the webpage and select inspect. Then you want to click on the console tab. From here you will see information pertaining to your skills running or crashing live. Look at the above photo. When the skill Dancing runs it first calls a command ‘Debug’ which displays the string value “The Dancing skill is starting!” (Keep in mind that debug command only executed first because it was called first. You aren’t required to call these debug commands, but they can be incredibly helpful with the traversal of your code). Then you can see each command following the debug message concerns Misty doing something which is simulating a dance. We suggest running code samples to practice running a skill.

7. You may stop running skills at anytime through the manager by clicking stop on the skill itself or the stop all skills button.



As you can see, the console displays information regarding the successful ending of the Dancing skill. Fatal errors will also be seen here highlighted in red. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Once you are finished coding/debugging, be sure to disconnect from Misty through the website. Turn the Misty robot off before placing her back in the container.

This manual is a basic outline of how to get started incorporating and developing your own skills for Misty! Proceed to the technical manual to get better acquainted with the dashboard, skills, development methodologies, coding standards, and diagrams specifically designed and implemented by the Bodacious Battle Bots.